Claims:

What is claimed is:

1. A system for performing adaptive optimization of application code within a virtual machine environment that uses a run-time compiler to compile application code for use by the virtual machine, the system comprising;

an optimizer which utilizes optimization parameters to calculate a dynamic size metric for the application code, and which size metric is then used to perform application code optimizations during run-time; and,

a feedback mechanism which uses information from the run-time compiler to determine at least some of said optimization parameters.

2. The system of claim 1 wherein the virtual machine is a run-time system.

3. The system of claim 2 wherein the virtual machine is a Java Virtual Machine.

4. The system of claim 1 wherein some of said optimization parameters may also be specified by a software developer prior to run-time.

5. The system of claim 1 wherein each of the determined factors 1 though N are assigned weights prior to being analyzed by the optimizer.

6. The system of claim 5 wherein each of the determined factors are assigned weights statically.

7. The system of claim 5 wherein each of the determined factors are assigned weights based on runtime information.

8. The system of claim1 wherein the factors are used to vary the operation of the optimizer from one system or platform to another.

9. The system of claim 1 wherein factors that have a substantial effect on the system performance are weighted accordingly for use in providing the system performance feedback.

10. The system of claim 1 wherein the size metric is evaluated by the formula

$$S(code) = \left(\sqrt[n]{\prod_{1...n} C_n V_n} \right)^x$$

wherein v represents variables used in evaluating the size matrix, c represents weights associated with each variable, and x represents a growth factor exponent.

11. A method for performing adaptive optimization of application code within a virtual machine environment that uses a run-time compiler to compile application code for use by the virtual machine, the method comprising the steps of:

gathering information about an application code and optimization parameters during run-time;

passing said information via a feedback mechanism to an optimizer;

calculating a dynamic size metric for the current application code using

said optimization parameters; and,

optimizing the application code based on the dynamic size metric.

12. The method of claim 11 wherein the virtual machine is a run-time system.

13. The method of claim 12 wherein the virtual machine is a Java Virtual Machine.

14. The method of claim 11 wherein some of said optimization parameters may also be specified by a software developer prior to run-time.

15. The method of claim11 wherein each of the determined factors 1 though N are assigned weights prior to being analyzed by the optimizer.

16. The method of claim 15 wherein each of the determined factors are assigned weights statically.

17. The method of claim 15 wherein each of the determined factors are assigned weights based on runtime information.

18. The method of claim 11 wherein the factors are used to vary the operation of the optimizer from one system or platform to another.

19. The method of claim 11 wherein factors that have a substantial effect on the system performance are weighted accordingly for use in providing the system performance feedback.

20. The method of claim 11 wherein the size metric is evaluated by the formula

$$S(code) = \left(\sqrt[n]{\prod_{1...n} C_n V_n} \right)^x$$

wherein v represents variables used in evaluating the size matrix, c represents weights associated with each variable, and x represents a growth factor exponent.